Crossing the Rubicon: A Generic Intelligent Advisor

Răzvan Andonie

J. Edward Russo

Rishi Dean

Crossing the Rubicon: A Generic Intelligent Advisor

Răzvan Andonie, J. Edward Russo, Rishi Dean

Abstract: Recommender systems (RS) are being used by an increasing number of e-commerce sites to help consumers find the personally best products. We define here the criteria that a RS should satisfy, drawing on concepts from behavioral science, computational intelligence, and data mining. We present our conclusions from building the WiseUncle RS and give its general description. Rather than being an advisor for a particular application, WiseUncle is a generic RS, a platform for generating application-specific advisors.

Keywords: Recommender systems, electronic commerce, user interface, user modeling.

1 Introduction

E-commerce sites use RS to guide visitors through the buying process by providing customized information and product recommendations. Some actual online recommender systems are described in [26, 31, 32]. Several well-known e-commerce businesses use, or have used, RS technology on their web sites: Amazon, Travelocity, BMW, MovieFinder, and Dell among them. Although commercial RS use began several years ago, we are still only beginning to use such systems on a large scale. Overviews of the relatively short history of RS and the techniques used may be found in [16, 30].

What is an "intelligent RS"? We will consider as intelligence the use of artificial intelligence features, such as adaptation, integration of learning algorithms, explanation, and case-based planning. Such an intelligent product search engine for online catalog sales is Analog Devices [1], developed at University of Kaiserslautern.

We begin by specifying the performance goals of a RS. This delineation of what a successful RS needs to be able to do leads to an analysis of the criteria that a successful RS must meet. Then we describe our own RS, named WiseUncle. The paper concludes with the results of the preliminary tests of WiseUncle performed by three e-commerce businesses. A preliminary paper describing our results may be found in [2].

2 Goals

The most telling sign of a RS's success is that it is used. On the one hand, the seller finds enough benefits to mount it on the company's website. On the other hand, site visitors find it both intelligent and trustworthy enough to purchase its recommended product. Both consumers and seller have goals that a RS must help them achieve. Many of these goals are closely linked because unless the consumer is satisfied by the interaction with the RS, a sale will not be made and the RS's main benefit to the seller is lost. Thus, the seller must provide a RS that meets the consumer's goals.

2.1 Consumer Goals

Consumers vary greatly in their product knowledge. Yet no matter how little they know and how few of the RS's queries they can respond to, they want an accurate, believable, and customized recommendation of the product they should buy. The overarching task of any RS is to start with the knowledge a consumer can provide and end with a recommendation of the best product tailored to that particular individual. Said differently, a RS bridges the gap between what consumers know and the one fact that

they want to know, the single best product for them. To accomplish this task, the RS must be built with knowledge from experts about both consumers and products and from mining the data of past purchases. Even after all this knowledge has been embedded in a RS, the resulting recommendation must not only be accurate, but also credible to the potential buyer. It will do neither buyer nor seller any good if the former doesn't believe the recommendation, even one that is truly insightful and fully matches the buyer's needs.

2.2 Seller Goals

Sellers also want the RS to deliver an accurate recommendation. (One reason, infrequently acknowledged, is the need to minimize exposure to lawsuits by buyers who feel that they were misled by an incompetent RS.) Because the RS is mounted on a website, sellers need it to function independently of any human assistance (sometimes just because humans are expensive). This desired independence requires that the RS must deal with whatever information the consumer can provide, however sparse or inconsistent. It must also function in real time. And it must do all this for the complete set of (possibly) millions of available products, as for automobiles, personal computers, multi-location vacations, and other products constructed from components. For a related kind of large problem for a RS, see [15].

Finally, sellers have goals beyond an immediate purchase. They also want a RS to help improve the long-term business relationship. Thus, an ideal RS should facilitate site visitors' return to the website (even if they did not purchase), incite customers to recommend the website to acquaintances, and build a positive image for the company. A technologically sophisticated and customer-friendly website is naturally associated with a technologically sophisticated and customer-friendly company. In addition, a RS can bring valuable information to the company's marketers, enabling them to improve their products, prices and promotions. For instance, through customer profiling or market segmentation, a RS can help a business decide to whom to send a customized offer or promotion.

3 Criteria

In this section we list the main criteria that a RS must satisfy. We partition them into two parts, those relating to system design and those more pertinent to the user "conversation". However, because the two are linked, our distinctions are necessarily imprecise. A different set of criteria can be found in [12].

3.1 System design criteria

Knowledgeable

A successful RS must utilize both global and local knowledge of both products and customers. By global customer knowledge we mean knowledge that is independent of the particular product domain, such as knowing how people make purchase decisions or how best to "converse" with them online. Global product knowledge includes the product's attributes, their functionalities, and anything else that is independent of the particular seller's offerings. A domain expert in personal computers, for instance, knows video cards, their performance capabilities (e.g., which type of card is needed for displaying photographs, playing video games, or showing movies), their approximate prices, and even how rapidly their technology is changing. Local customer knowledge refers to the ability to link a customer's personal needs, uses, and preferences to the focal product. Thus, a consumer may have a high need for status among coworkers that might influence many purchases. However, local consumer knowledge enables a RS to direct the consumer toward a particular product, such as a personal computer with the image of the latest and greatest technology or an automobile that signals status (at least to the target audience of coworkers). Local product knowledge focuses on the vendor's offerings. Such knowledge not only

includes models, styles, components, add-ons, etc., but also which components can be configured with others and the moment-to-moment availability of any recommended product or special offers. Thus, a RS should be connected to the vendor's product database and to the pertinent marketing reporting systems. The volatile nature of both real-world products like personal computers and vendors' information systems requires adequate database maintenance in the RS. Finally, a successful RS must not only integrate all four types of knowledge, but it must make all of this knowledge visible to the user. That is, this knowledge should not only be "under the hood" of a good recommender system, but users must be able to see how the full array of knowledge is used and why the resulting recommendation has been intelligently customized to their personal needs.

Viable RS can be constructed from less than the full array of sources of knowledge just described. However, some functionality will be sacrificed. For instance, global consumer knowledge is used by RS that require users to input numerical estimates like the importance weights of the main product attributes. Such systems more or less ignore the decision process that consumers prefer to use (i.e., a part of global consumer knowledge). More extremely, a RS can be built with essentially no expert knowledge relying on data mining alone. Using only a record of past products purchased and the assignment of consumers to market segments, a RS can recommend products that were sold successfully in the past to each segment, a kind of popularity poll based on market share. However, the customization to a single consumer is likely to be crude. Note that recommendations based on expert knowledge are much better at dealing with new products, whereas market share rankings can only be used for products that have been on the market and changed little over time.

Customization

The task of making a customized recommendation requires the RS to know how to draw multiobjective comparisons among products. This is a classic challenge for any decision maker, human or software. It is made more complex for a RS because the information extracted from a potential purchaser is limited compared to that needed for at least some ideal solutions [13, 6, 11]. Nonetheless, the RS must somehow trade off the relative importance of different product attributes or features in order to achieve the best customization. Many automobile buyers face a difficult tradeoff between gas mileage and safety. The former points toward smaller and lighter vehicles, while the latter is associated with larger and heavier ones.

Scalability and real-time performance

In order to meet the requirements of an Internet application, a RS should be scalable and able to work in real-time, even for very large problem sizes. For instance, a recommender system connected to a large web site must produce each recommendation within a few tens of milliseconds while serving thousands of consumers simultaneously and searching through potentially millions of possible product configurations.

Imperfect data

A RS should be robust in the face of data that are uncertain, noisy, sparse, or missing. One source of value in an intelligent RS lies in the fact that most customers have not deeply considered many of the available products and their product features, or are unable to recognize and express all of their personal needs related to the product. However, the downside to this accommodation to users' limited knowledge is that we must often deal with sparse or missing data that result from a customer responding, "I don't know", "It doesn't matter to me", or "I don't want to answer this question". In addition, the less consumers know about the product, about their personal needs, and especially about the links between them, the more often the RS will have to resolve inconsistent input.

Connection to database

RS should be connected to the vendor's product database and the marketer's reporting systems. Only products that can be configured in a feasible manner should be recommended. And of these legitimate products, only those that are currently in stock should be recommended. The highly volatile nature of real-world products and information systems requires adequate database maintenance in the RS.

Retention of customer data

During each interaction with a customer, the RS extracts knowledge from the customer that should be used to build and update a customer profile. The RS should save this profile so that it, and aggregations of profiles, can be "mined" for marketing-relevant knowledge. That knowledge may be as straightforward as a record of products purchased by different customer segments. Alternatively, it may be as complex as inferring underserved segments in a space of consumers. It may even include the results of deliberate experimentation with, for example, the bundling of components, price discounts, complementary products, and so forth. The point that is relevant to a successful RS is that it must be able to retain enough of the knowledge that it gains, such as customer profiles, to be able to provide useful answers to a range of questions asked by marketing and product managers.

Learning

The RS should be able to improve its functionality by continually learning from its interactions with consumers. This learning can be thought as reward and penalty feedback. After selling a recommended product, the RS will enhance that type of recommendation for customers with similar profiles. When the recommended product is not purchased, the RS will make that recommendation less probable for the next similar customers. This learning depends the availability of sales data.

Another type of interactive learning comes from the conversation with the customer. If customers customers find the conversation too long or uninteresting, they will not complete the recommendation process and force a partial recommendation. The RS should adapt to this situation and improve future conversations with similar customers.

Domain independence

Ideally, a RS should be largely domain-independent, so that with minimum modification, one should be able to customize the same platform for other applications (e.g., selling computers, cars, financial services, or vacations). Inter alia, this means that the system design should clearly separate the generic part from the domain-specific knowledge modules.

3.2   User conversation criteria

The RS's customer interface should be based on the psychology of the consumer and the purchase decision process [14]. Therefore, behavioral science techniques should guide the customer dialog. That conversation with the user should be based on how consumers actually think, rather than forcing them to feed an optimization algorithm. By speaking the language of the user, a successful RS provides welcome relief to those users from some of the complexity of the purchase decision.

The customer-recommender interface is usually based on a series of interactive questions presented to the customer by the RS, accompanied by multiple-choice options for the customer to input their answers. For an alternative conversational structure, see [10]. One task is to devise an optimal strategy to select the next question to be posed to the user. An intelligent dialog should be personalized, with a subsequent question based on the responses to all prior ones. That still leaves the challenge of determining what exactly is meant by optimal. One reasonable proposal is to minimize customer effort, typically measured

as the duration of the dialog, while still being able to make accurate predictions [34, 35, 36]. However, this strategy may be based on too simple a behavioral criterion. We believe that the standard should not be customer effort only, but overall customer satisfaction. The quantification of overall satisfaction must be derived from longer-term statistics on system usage and surveys of customers. One twist on the optimization of question sequence is inserting a small amount of randomness when selecting the questions. This may help to extend the space over which the recommender understands the customer's interests and ensures that all questions are occasionally presented to customers.

During the conversation, the WiseUncle RS adopts a five-stage process for making the purchase decision described by Horowitz and Russo [25]. The stages are: Opening; Utilitarian Needs; Hedonic Preferences; Optional Features or Add-ons; and Endgame.

Stage 1, Opening, frames the buyer (e.g., the buyer's level of knowledge of the product category and extent of product search to date). It also identifies the buyer's preference among gross product characteristics (e.g., a desktop versus a laptop personal computer or a one-week versus a two-week cruise). Stages 2 and 3 encompass, respectively, the utilitarian and hedonic or emotional needs [3]. The former include the functional uses of the product, such as an automobile's seating capacity or environmental friendliness or whether a personal computer is going to be used by teenagers or for showing movies. Stage 3's hedonic needs, like the image of a car's brand name or body style, are often harder for a buyer to express. Thus, it can be a substantial behavioral challenge not only to use such knowledge to identify the best product but also to extract hedonic knowledge from the consumer. Stage 4, Optional Features or Add-ons, captures the remaining, minor product specifications, like an automobile's audio system or aspects of its interior. The final stage, Endgame, covers such external elements as the local availability of reliable repair service for an automobile, a PC's warranty, or travel insurance for a vacation cruise. These five stages are sufficient to structure the process of a purchase decision for even the most complex products.

The following factors contribute to the success of such conversations in Internet-mediated dialogs [24]. They form the remaining criteria in the design of a RS.

The benefits of a conversation exceed its costs

People use information only if it is perceived as adding benefits or as reducing costs. If (expected) costs exceed (expected) benefits at any point, there is a clear risk of the customer terminating the dialog. Thus, a criterion for a successful conversation is site visitors continually receiving some benefit during, not only after, the interactions/dialog.

Credibility and trust

Information and advice must be credible, and the source must be trustworthy [7, 8, 5]. An Internet-delivered RS cannot provide the face-to-face cues of trustworthiness that a human can. However, although a RS may have no initial reputation for trust (based on past experience), such an image can be built over time by personal usage, word-of-mouth recommendations, or public endorsements (e.g., the endorsement of the system's knowledge and disinterestedness in consumer-oriented columns in newspapers and magazines). One alternative is to add a confidence metric (like the Average Customer Rating used by Amazon), and this has the potential to improve user satisfaction and alter user behavior in the RS [27]. A second alternative is to make the RS adaptive. This would reduce the risk of manipulation: users can detect systems that manipulate their predictions, something that that has a negative impact on their trust [28].

Intelligence

The RS must display intelligence in the conversation. For instance, it must know what kinds of information people can validly provide and how to successfully extract that information from buyers. Consumers can usually say what they need or want the product to do and can often articulate such personal preferences as style and color. However, they may have difficulty specifying the product features that meet those needs. Intelligence can also be demonstrated by prompting users about needs that might be overlooked. Such needs may include those of others (family members who will use the new personal computer), future needs (an automobile that suits the needs of an elderly parent who may visit regularly), and unanticipated needs (especially first time product users like vacationers to an "adventure" destination). Intelligence is manifest in many small ways, such as clearly remembering the answer from a previous query and incorporating it into a subsequent question. It can also appear as the rephrasing of questions to reflect the consumer's level of knowledge in the product domain.

Control

Customers should be able to request additional or explanatory information. Or as the conversation proceeds, the customer may learn something that requires returning to an earlier point in the dialog and changing a preference stated there. Buyers who feel impatient should be able to request a recommendation at any time, even before the RS would normally feel comfortable providing one. Finally, the buyer might even like to suspend the conversation and return later. More control in any situation is empowering, and more so in situations where control is expected. Providing satisfactory conversational control is a special challenge to RS.

Feedback

A desire for feedback in a conversation is natural, so appropriate kinds enhance user acceptance of a dialog with a RS. Specific feedback might include (a) how much progress has been made toward identifying the best product, and (b) how much longer the conversation is expected to take. Whatever specific feedback options are provided, however, users do not want to receive feedback only after they have answered every question (as they must in many static surveys).

How might we know that a RS fully meets all of the above criteria? That question has two answers, a real and an ideal. The real answer is that it depends on customer acceptance. This is, we can try to build into the RS all of the desired capabilities, but only users can validate success. We may believe that our system is intelligent or provides plenty of feedback, but if its users believe that the system has failed to do so, then it has. Working in the RS's favor is that users often judge a system's performance relative to that of the other systems they know. They do not expect the WiseUncle system described below to be as good as a genuine human wise uncle, someone who is very knowledgeable and is on their side. Thus, superior performance rather than perfection may be sufficient for success.

The ideal criterion is a kind of Turing test for RS. That is, ideally the potential buyers would not be able to determine whether they were interacting with a RS or with a (very fast) human advisor who was knowledgeable and on their side (sometimes called a human concierge [9]). This ideal is not as impossible to achieve as it may first sound. No human advisor is perfect, so the RS doesn't have to be either. We do not expect this ideal goal to be achieved soon by any RS. However, it may be worth keeping in mind as an ideal for system designers.

4 System description

Rubicon is a generic domain-independent advisor, recommending products from an existing set. Each product is configurable, meaning it is comprised of several components, which may each be de-

scribed in turn by several attributes. Building a RS depends largely on the knowledge representation model, and we chose a computational intelligence framework. Our RS is a classifier that "learns" to make good recommendations. This classifier is an expert system, able to explicitly expose its acquired knowledge. The main characteristics of Rubicon are:

- The inferential process from the customer's needs to the best product is constructed in two stages, called *Bridges*, one from needs to product attributes, and the second from attributes to the products themselves.

- It can easily be customized for different applications since the interface to the application-specific knowledge domain is separated from the main system.

- The front-end dialog is dynamic and changes based on user responses. It elicits information from users to determine a location in a needs space which is then used to find optimal (sub-optimal) solutions in a products space.

- It accepts imprecise input from users.

- It provides a justification for all recommendations.

- Reversibility: The system can reverse the decision process from effect to cause. This allows forecasting the adoption of new products or services using real customer decision data.
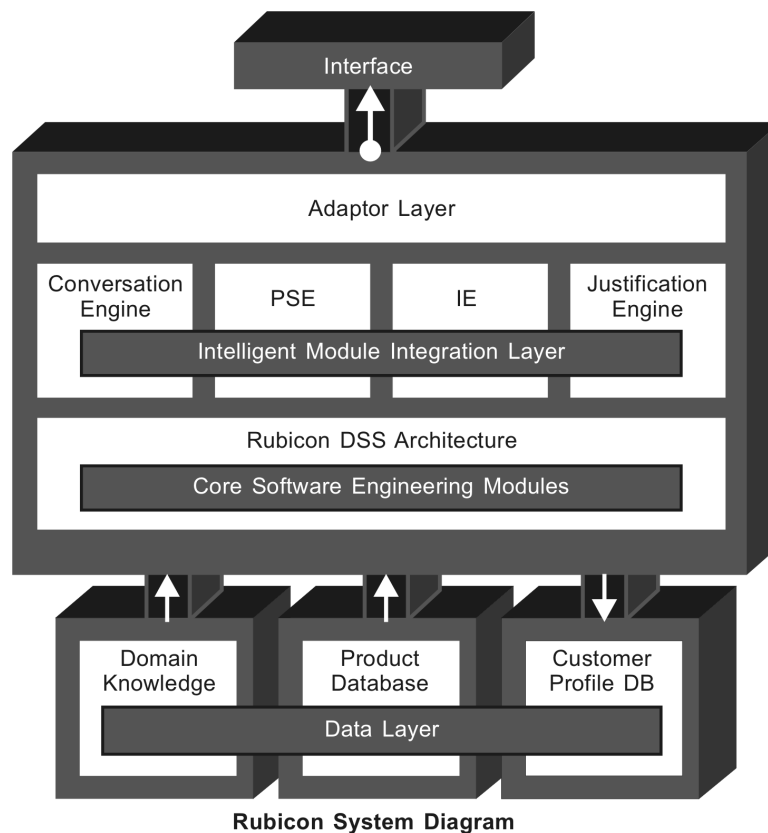

Figure 1: Rubicon High Level System Description

The Rubicon system diagram (Fig. 1) shows the following main modules.

4.1　Conversation Engine (CE)

The CE is responsible for dialog management, presenting questions to the user and processing the resulting responses sent to it by the user. Questions and their associated responses are processed to accomplish the following two results: *i)* Propagate the knowledge gained from a response to the subsequent inference mechanisms and *ii)* Determine the next question to pose to the user.

In doing so, the dialog management occurs subject to the following constraints:

- Presents the appropriate questions for the system to confidently determine an intelligent, personalized recommendation.

- Presents questions conforming to users' expectation of a real dialog with respect to flow, organization, and coherence.

- Minimizes the number of questions presented.

- It enables scalable addition/ subtraction/ modification of questions.

- Allows users and administrators to reproduce particular dialogs.

- Uses proper constructs for further data mining.

From a computational standpoint a rule-based expert system is used to implement the CE's dialog management process. Questions and responses are linked by sets of predetermined rules, and a number of other intermediary constructs. In this way, the questions, responses, and rules can be specified, along with goals (i.e., knowledge to be gathered) independently of knowing the dialog flow in advance. The system at runtime determines, based on the behavioral and informational goals, which question to present next to the user. When all appropriate questions have been presented, the conversation is determined to be complete. However, the user may intervene at any time to ask for the system's current best recommendation based on the information provided thus far.

4.2　Inference Engine (IE)

The purpose of the IE is to map the user's profile of needs (the output of the CE) to the attributes necessary to comprise the appropriate recommendation. Given a set of responses resulting from the dialog, the IE can indicate a set of recommendations, ordered by the degree of their preference. These recommendations are not concrete (physical) product recommendations yet, but a mapping from the user needs space to the space of attributes, yielding generic descriptions of the product, like "RAM Amount" (e.g., standard, large, maximum) and "Network Card type". Collectively this inference is called the *First Bridge*.

The IE is taught by a human expert. However, it can learn incrementally as well: new teaching examples can be added without restarting the teaching process from the beginning. Conditional rules can be extracted to describe the behavior of the IE and justify recommendations, market research and performance improvement. The IE is stable under noisy inputs and user uncertainty. Such "noise" may be produced by "I don't know" answers, or by contradictory answers in the dialog.

A neuro-fuzzy network incorporates fuzzy, rather than crisp, membership functions in its structure and has the ability to learn when presented with training data [33]. After training, such a network can be used as a classifier. The fuzzy neural network classifier builds decision boundaries by creating subsets of the pattern space. It is a model-free estimator and does not make assumptions of how outputs depend on inputs. Instead, it adjusts itself to a given training set by learning algorithms and decides the boundaries of classes. When given an unknown pattern, the fuzzy inference network classifier uses the learned

knowledge to estimate the membership value of this pattern in each class and classify the input pattern according to the membership values.

To implement the IE, we have used the fuzzy neural net architecture introduced in [33], trained to represent the expert knowledge of a particular product domain. For instance, in the case of a personal computer RS, experts develop training patterns to represent the varying needs profiles of customers along with their corresponding feature sets for a recommended PC. The inference process is fast, online.

4.3 Product Search Engine (PSE)

The PSE is the *Second Bridge*, a mapping from the space of attributes to the space of (physical) products. It is an optimization module interfacing with the retailer's product database to select the best, valid product configurations that match the criteria specified by the user, such as the minimum cost, the maximum likelihood of success, or a number of other simultaneous criteria. The inputs to the PSE are the levels of the attributes (the output of the IE), the configuration constraints (i.e., incompatibilities among components), and a user's criteria for optimization (e.g., a desired price point). These criteria for the algorithm can be set by the IE and CE and are, therefore, uniquely tailored to a given user. The PSE can sort though billions of options in real time, allowing searches to be completed online. The products with the highest degree of fit are passed to the Justification Engine for further processing.

The response to a question is subsequently used to provide more information that adds to Rubicon's knowledge base during a user-experience. This, in turn, leads to a recalculation and optimal selection of the next most appropriate question. This question-response model continues until Rubicon is either asked, or has sufficient confidence, to make a recommendation.

The PSE navigates a vast search space, taking into account different optimization criteria. We used a genetic algorithm approach for this (NP-complete) optimization problem. In the initial phases, the PSE operates on abstractions of the real world, and then through an adaptor layer translates these abstractions into concrete items. This information is capable of being read and processed at runtime. The PSE remains independent of constant updating of the "real world" items. This adaptor level is implemented as an XML data bridge.

4.4 Justification Engine (JE)

Recommendations are run through the JE to provide a plain English explanation of why the system has provided a specific recommendation. This justification is delivered in the same vernacular as the dialog, personalized to the user, and is present to facilitate user understanding and adoption of the recommendation. This can be accomplished by transforming the IEŠs mathematical equations into linguistic rules more easily understood by the human.

A nice feature of the fuzzy-neural architecture used in the IE module is that it can be expressed by a set of fuzzy IF/THEN inference rules and these rules can be easily extracted automatically [33]. The JE takes the set of IF/THEN rules from the IE and the set of recommended configurations from the PSE and develops a rationale for selecting each product. The value of the JE is that it creates confidence in the recommendation.

5 Preliminary tests

Rubicon is implemented using a complimentary modular software approach that encapsulates the individual computational blocks, as well as the necessary software architecture emphasizing a stable and reusable model that is compliant with the J2EE technology standard.

Although still under development, Rubicon was sufficiently developed to be submitted to usability testing by two major PC manufacturers. Each test involved about a dozen users and compared three RS.

One was the manufacturer's current online RS, one was an attractive competitor, while the third was Rubicon. The results made available revealed that Rubicon was judged clearly superior in both tests. For instance, in one test, when asked which of the three RS the user would "be most likely to use again", nine of eleven respondents chose Rubicon.

Rubicon was tested online by a webhosting services provider. Of 2200 online users who began a conversation, 83% completed it to the point of receiving a recommendation (which was the only result made available to us). This was judged by the host company to be an extraordinary high completion (i.e., non-abandonment) rate.

6   Conclusions

We have built Rubicon to meet the criteria described in Section 3. We have used principles and techniques from artificial intelligence and behavioral sciences. Since we have focused on the core system, other modules of Rubicon, used for prediction, customer profiling, and marketing segmentation were omitted. It was a challenging task to build Rubicon, especially because of its generic character. Making the system largely independent of a specific e-commerce application required greater complexity and abstraction. But do we really need a generic RS? From a user perspective this may be a non-issue. However, for the RS designer and software engineer this is a critical requirement. We should think not only in terms of how to use a RS, but also how to build it and how to adapt it fast for very different application areas.

References

[1] I. Vollrath, W. Wilke, and R. Bergmann, Case-based reasoning support for online catalog sales, *IEEE Internet Computing*, July-August, pp. 47-54, 1998.

[2] R. Andonie, J. E. Russo, and R. Dean, Crossing the rubicon for an intelligent advisor, *Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05*, San Diego, CA, pp. 7-12, 2005.

[3] R. Batra and O. T. Ahtola, Measuring the hedonic and utilitarian sources of consumer attitudes, *Marketing Letters*, Vol. 2, pp. 159-170, 1990.

[4] M. Bilgic and R. J. Mooney, Explaining recommendations: Satisfaction vs. promotion, *Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05*, San Diego, CA, pp. 13-18, 2005.

[5] P. Bonhard, Who do trust? Combining recommender systems and social networking for better advice, *Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05*, San Diego, CA, pp. 89-90, 2005.

[6] G. Carenini, User-specific decision-theoretic accuracy metrics for collaborative filtering, *Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05*, San Diego, CA, pp. 26-30, 2005.

[7] V. F. Kleist, A transaction cost model of electronic trust: Transactional return, incentives for network Security and optimal risk in the digital economy, *Electronic Commerce Research*, Vol. 4, 41-57, 2004.

[8] M. A. Patton and A. Josang, Technologies for trust in electronic commerce, *Electronic Commerce Research*, Vol. 4, pp. 9-21, 2004.

[9] M. J. Pazzani, Beyond idiot savants: Recommendations and common sense, *Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05*, San Diego, CA, pp. 99-100, 2005.

[10] H. Sawamura, M. Yamashita, and Y. Umeda, Applying dialectic agents to argumentation in e-commerce, *Electronic Commerce Research*, Vol. 3, pp. 297-313, 2003.

[11] J. B. Schafer, DynamicLens: A dynamic user-interface for a meta-recommendation system, *Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05*, San Diego, CA, pp. 72-76, 2005.

[12] S. Spiekermann and C. Paraschiv, Motivating human-agent interaction: Transferring insights from behavioral marketing to interface design, *Electronic Commerce Research*, Vol. 2, pp. 255-285, 2002.

[13] G. Urban, F. Sultan, and W. Qualls, *Design and evaluation of a trust based advisor on the Internet*, Cambridge, MA, MIT Press, 1999.

[14] P. Warnestal, Modeling a dialogue strategy for personalized movie recommendations, *Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05*, San Diego, CA, pp. 77-82, 2005.

[15] T. Zhu, R. Greiner, G. Haubl, B. Price, and K. Jewell, Behavior-based recommender systems for web content, *Proceedings of the Wokshop Beyond Personalization 2005, in conjunction with the International Conference on Intelligent User Interfaces IUI'05*, San Diego, CA, pp. 83-88, 2005.

[16] W. Lin, S. A. Alvarez, and C. Ruiz, Efficient adaptive-support association rule mining for recommender Systems, *Data Mining and Knowledge Discovery*, Vol. 6, pp. 83-105, 2002.

[17] J. B. Schafer, J. Konstant, and J. Riedl, Electronic commerce recommender applications, *Data Mining and Knowledge Discovery* 5, pp. 115-152, 2001.

[18] C. Basu, C., H. Hirsh, and W. Cohen, Recommendation as classification: Using social and content-based information in recommendation, *Proceedings of the 1998 National Conference on Artificial Intelligence (AAAI-98)*, AAAI Press, pp. 714-720, 1998.

[19] N. Good, J. B. Schafer, J.A. Konstan, A. Borchers, B. Sarwarand, J. Herlocker, and J. Riedl, Combining collaborative filtering with personal agents for better recommendations, *Proceedings of the 1999 National Conference on Artificial Intelligence (AAAI-99)*, AAAI Press, pp. 439-446, 1999.

[20] P. Resnick and H. R. Varian (eds.), Recommender systems, *Special issue of Communications of the ACM*, Vol. 40, 1997.

[21] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, Analysis of recommendation algorithms for e-commerce, *Proceedings of the ACM E-Commerce Conference*, pp. 158-167, 2000.

[22] A. M. Rashid, I. Albert, D. Cosley, S. K. Lam, S. McNee, J. A. Konstan, and J. Riedl, Getting to know you: Learning new user preferences in recommender systems, *Proceedings of the 2002 International Conference on Intelligent User Interfaces*, San Francisco, pp. 127-134, 2002.

[23] J. Breese, D. Heckerman, and C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, Madison, WI, 1998.

[24] J. E. Russo, Aiding purchase decisions on the Internet, *Proceedings of the Winter 2002 SSGRR (Scuola Superiore Guglielmo Reiss Romoli) International Conference on Advances in Infrastructure for Electronic Business, Education, Science, and Medicine on the Internet*, L'Aquila, Italy, 2002.

[25] A. Horowitz and J. R. Russo, Modeling new car customer-salesperson interaction for a knowledge-based system, *Advances in Consumer Research*, Vol. 16, pp. 392-398, 1989.

[26] J. Liu and J. You, Smart shopper: An agent-based web-mining approach to Internet shopping, *IEEE Transactions on Fuzzy Systems*, Vol. 11, pp. 226-237, 2003.

[27] S. M. McNee, S. K. Lam, C. Guetzlaff, J. A. Konstan, and J. Riedl, Confidence displays and training in recommender systems, *Proceedings of INTERACT '03 IFIP TC13 International Conference on Human-Computer Interaction*, pp. 176-183, 2003.

[28] D. Cosley, S. K. Lam, I. Albert, J. Konstan, and J. Riedl, Is seeing believing? How recommender systems influence users' opinions, *Proceedings of CHI 2003 Conference on Human Factors in Computing Systems*, pp. 585-592, Fort Lauderdale, FL, 2003.

[29] L. M. Rocha, TalkMine: A soft computing approach to adaptive knowledge recommendation, *Soft Computing Agents: New Trends for Designing Autonomous Systems - Studies in Fuzziness and Soft Computing*, V. Loia and S. Sessa (eds.), Physica-Verlag, Springer, pp. 89-116, 2001.

[30] J. L. Herlocker, J. Konstan, L. G. Terveen, and J. Riedl, Evaluating collaborative filtering recommender systems, *ACM Trans. Inf. Syst.*, Vol. 22, pp. 5-53, 2004.

[31] S. Y. Hwang, W. C. Hsiung, and W. S. Yang, A prototype WWW literature recommendation system for digital libraries, *Online Information Review*, Vol. 27, pp. 169-182, 2003

[32] H. W. Tung and V. W. Soo, A personalized restaurant recommender agent for mobile e-service, *Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, pp. 259-262, 2004.

[33] L. Y. Cai and H. K. Kwan, Fuzzy classification using fuzzy inference networks, *IEEE Transactions on System, Man, and Cybernetics, Part B: Cybernetics*, Vol. 28, pp. 334-347, 1998.

[34] M. Doyle and P. Cunningham, A dynamic approach to reducing dialog in online decision guides, *Advances in case-based reasoning, Proc. of the 5-th European Workshop, EWCBR-2000, Trento, Italy, September 6-9, 2000*, Springer, pp. 49-60, 2000.

[35] S. Schmitt and R. Bergmann, A formal approach to dialogs with online customers, *Proc. of the 14-th Bled Electronic Commerce Conference, Bled, Slovenia, June 25-26, 2001*, pp. 309-328, 2001.

[36] D. McSherry, Minimizing dialog length in interactive case-based reasoning, *Proceedings of the 17-th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pp. 993-998, 2001.

Răzvan Andonie

Computer Science Depart., Central Washington University, Ellensburg, USA, andonie@cwu.edu

J. Edward Russo

Johnson Graduate School of Management, Cornell University, Ithaca, USA, jer9@cornell.edu

Rishi Dean

Visible Measures Corp., Cambridge, MA, USA, rdean@visiblemeasures.com